

Pengaturan Gerakan *Hover* dan *Roll* pada *Quadcopter* dengan Menggunakan Metode PI Ziegler-Nichols dan PID Tyreus-Luyben

Huda Ubaya¹, Bambang Tutuko², Borisman Richardson³, Sutrimo⁴

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya
Jl. Raya Palembang Prabumulih KM. 32 Inderalaya, Ogan Ilir, Sumatera Selatan

¹huda_ubaya@yahoo.com, ²beng_tutuko@yahoo.com, ³borismanrichardson@gmail.com, ⁴slamet.sutrimo@gmail.com

Intisari—*Quadcopter* merupakan jenis robot terbang yang bergerak dengan menggunakan empat buah motor dan memiliki empat gerakan dasar. Diantara gerakan tersebut adalah gerakan *hover* dan gerakan *roll*. Gerakan *hover* dipengaruhi oleh kecepatan putar seluruh motor secara bersamaan sehingga menghasilkan gerakan berpindah tempat dari bawah ke atas dan sebaliknya sedangkan gerakan *roll* adalah gerakan rotasi pada sumbu *x* *quadcopter*. Agar *quadcopter* dapat melakukan pergerakan tersebut dengan baik, dibutuhkan pengendali. Pengendali yang digunakan pada penelitian ini adalah pengendali PI dan PID. Pengendali PID terdiri dari *proportional gain* K_p , *integral gain* K_i dan *derivative gain* K_d . Untuk menentukan nilai *proportional gain* K_p , *integral gain* K_i , dan *derivative gain* K_d , pada penelitian ini digunakan metode *tuning* Ziegler-Nichols untuk *hover* dan Tyreus-Luyben untuk *roll*. Didapatkan nilai pengendali dengan nilai *proportional gain* K_p sebesar 5,66914408 dan nilai *integral gain* K_i sebesar 0,05281733959 untuk gerakan *hover* sehingga menghasilkan grafik respon transient dengan overshoot 16%, rise time 3,75 detik, dan settling time 8 detik. Sedangkan untuk gerakan *roll*, nilai *proportional gain* K_p sebesar 67,11547084, nilai *integral gain* K_i sebesar 7,6267045, dan nilai *derivative gain* K_d sebesar 0,426113135.

Kata Kunci— *Quadcopter*, *hover*, *roll*, pengendali PID, Ziegler-Nichols, Tyreus-Luyben.

I. PENDAHULUAN

Unmanned Aerial Vehicle (UAV) atau pesawat tanpa awak, saat ini memiliki peranan dan aplikasi yang semakin berkembang, diantaranya dalam bidang militer, ilmu kebumihantaran untuk mengambil citra suatu wilayah, dan dibidang lainnya. Salah satu dari sekian banyak UAV adalah *multirotor*.

Multirotor adalah pengembangan dari *helicopter* yang dilengkapi dengan banyak motor, jumlah motor pada *multirotor* ada beberapa macam, untuk *multirotor* yang memiliki jumlah tiga motor disebut *tricopter*, jumlah empat

motor disebut dengan *quadcopter*, jumlah enam motor disebut dengan *hexacopter* dan *multirotor* dengan jumlah motor 8 disebut dengan *octacopter*.

Pada *Quadcopter* memiliki macam-macam mode atau susunan *frame*, yaitu kombinasi *frame* X, + (*plus*), dan V. [1]. Susunan *frame* tersebut dapat mempengaruhi gerak pada *quadcopter*, pada *frame* X dan *Plus* susunan *frame* tersebut bertujuan agar *quadcopter* lebih fokus pada kestabilan gerakannya, sedangkan model *frame quadcopter* V bertujuan pada kelincihan gerak *quadcopter*, atau lebih tepatnya *quadcopter* tipe *frame* V adalah tipe *frame* yang mengacu pada kecepatan gerak *quadcopter*.

Pada *quadcopter* memiliki gerakan-gerakan manuver yang diantaranya adalah gerakan *pitch*, *roll*, *yaw*, dan *hover*, dimana gerakan tersebut adalah gerakan dasar yang dapat dilakukan oleh *quadcopter*. Gerakan *pitch*, *roll* dan *yaw* adalah gerakan rotasi pada *quadcopter*, sedangkan *hover* adalah gerakan translasi pada *quadcopter*.

Gerakan-gerakan *quadcopter* diatur oleh *controller* PID. *Controller* PID berfungsi untuk mengatur kecepatan respon dan kestabilan gerakan *quadcopter*. untuk mendapatkan hasil yang diinginkan, maka inputan nilai PID pada *quadcopter* haruslah tepat. Komponen PID terdiri dari proporsional, integral, dan derivatif. Komponen proporsional, integral, dan derivatif dapat digunakan keseluruhan secara bersamaan maupun tersendiri tergantung pada respon yang diinginkan terhadap suatu *set-point* [2]. Untuk membangun sebuah pengendali, perlu pemodelan matematis *quadcopter* terlebih dahulu.

Ada beberapa metode penentuan nilai PID yang digunakan pada *quadcopter* diantaranya adalah metode Tyreus-Luyben, metode ini hampir sama dengan metode Ziegler-nichols,

metode tersebut mengharuskan mengenal plant motor terlebih dahulu, hanya saja yang membedakan metode ini dengan metode Ziegler-Nichols adalah pada tabel tuning rule-nya.

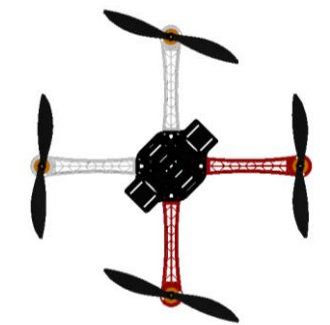
Adapun seperti penelitian sebelumnya [3] yang menggunakan metode PD Ziegler-Nichols untuk mengatur kecepatan respon dan kestabilan gerak rotasi *quadcopter*. Pada penelitian tugas akhir ini akan melakukan Pengaturan Gerakan *Roll* Pada *quadcopter* Dengan Menggunakan Metode PID Tyreus-Luyben agar *quadcopter* dapat melakukan gerak rotasi yang diinginkan, dan melakukan gerakan hover pada *quadcopter* dengan menggunakan metode Ziegler-Nichols.

II. METODOLOGI PENELITIAN

Dalam melakukan metodologi penelitian, dibutuhkan desain rangkaian *quadcopter* yang akan diteliti, kemudian akan dilakukan pemodelan terhadap desain tersebut sehingga dapat dirancang pengendali PI untuk mengendalikan gerakan *hover*.

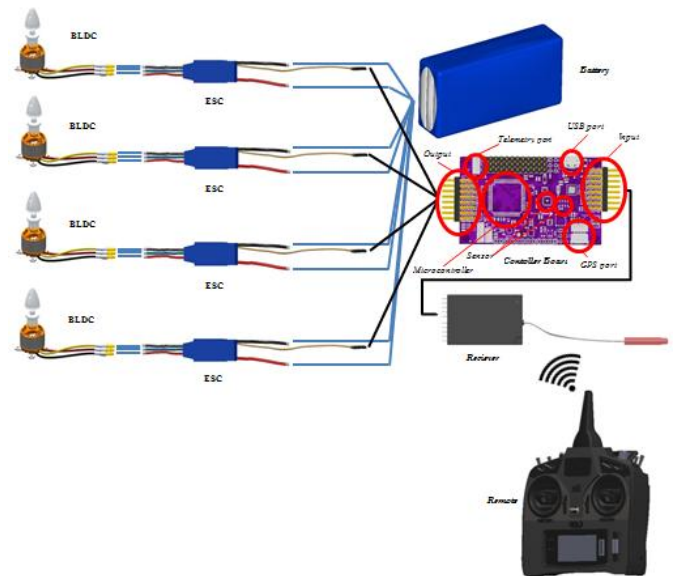
A. Desain Quadcopter

Desain *quadcopter* terdiri dari desain mekanik dan desain elektronik. Desain mekanik terdiri dari *frame* dan baling-baling *quadcopter* (Gbr. 1).



Gbr. 1. Bentuk *frame quadcopter*

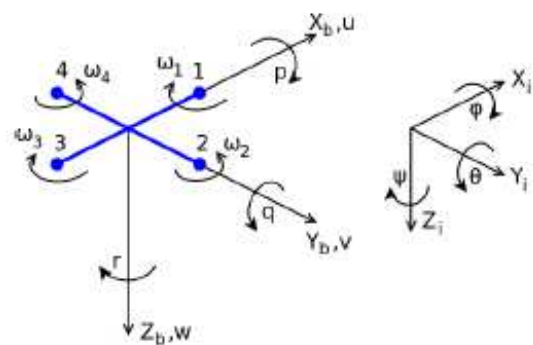
Desain elektronik merupakan rangkaian elektronik yang digunakan oleh *quadcopter* yang terdiri dari motor BLDC (*Brushless Direct current*), ESC (*Electronic Speed Controller*), *flight controller*, *battery*, dan *receiver-transmitter* (Gbr. 2).



Gbr. 2. Rangkaian *quadcopter*

B. Model Quadcopter

Pemodelan *quadcopter* dilakukan berdasarkan koordinat, orientasi, sudut, dan gaya-gaya yang bekerja padanya.



Gbr. 3. Sistem koordinat *quadcopter* [4]

Berdasarkan Gbr. 3, *quadcopter* memiliki tiga buah sumbu ortogonal yang masing-masing adalah sumbu X, Y, dan Z. Di setiap sumbu tersebut, *quadcopter* melakukan pergerakannya yang meliputi pergerakan kecepatan translasi $\vec{V}(t) = [u(t) \ v(t) \ w(t)]^T$ dan kecepatan rotasi $\vec{\Omega}(t) = [p(t) \ q(t) \ r(t)]^T$ dengan $u(t)$, $v(t)$, dan $w(t)$ masing-masing merupakan kecepatan translasi pada sumbu X, Y, dan Z, dan $p(t)$, $q(t)$, dan $r(t)$ masing-masing merupakan kecepatan rotasi pada sumbu X, Y, dan Z. Gerakan *hover* merupakan gerakan translasi $w(t)$ pada sumbu Z.

Dengan pengaruh Hukum Newton dan Efek Coriolis, kemudian dengan memodelkan persamaan kinematikanya

berdasarkan sudut Euler [5], *quadcopter* menghasilkan enam buah pergerakan dasar (1) – (6).

$$\dot{u}(t) = v(t)r(t) - w(t)q(t) - g \sin \theta(t) \quad (1)$$

$$\dot{v}(t) = w(t)p(t) - u(t)r(t) + g \sin \phi(t) \cos \theta(t) \quad (2)$$

$$\dot{w}(t) = u(t)q(t) - v(t)p(t) + g \cos \phi(t) \cos \theta(t) + \frac{\tau_z(t)}{m} \quad (3)$$

$$\dot{p}(t) = \frac{I_y - I_z}{I_x} q(t)r(t) + \frac{\tau_x(t)}{I_x} \quad (4)$$

$$\dot{q}(t) = \frac{I_z - I_x}{I_y} r(t)p(t) + \frac{\tau_y(t)}{I_y} \quad (5)$$

$$\dot{r}(t) = \frac{I_x - I_y}{I_z} p(t)q(t) + \frac{D_z(t)}{I_z} \quad (6)$$

Gerakan translasi dipengaruhi oleh gaya gravitasi g , sedangkan gerakan rotasi dipengaruhi oleh momen inersia I_x , I_y , dan I_z yang masing-masing merupakan momen inersia pada sumbu X , Y , dan Z . Gerakan translasi $\dot{w}(t)$ dihasilkan dengan memanfaatkan gaya *thrust* $\tau_z(t)$ (gaya *thrust* pada sumbu Z) yang dipengaruhi oleh massa *quadcopter* m . Dan gerakan rotasi $\dot{p}(t)$ dan $\dot{q}(t)$ dihasilkan dengan memanfaatkan gaya *thrust* $\tau_x(t)$ dan gaya *thrust* $\tau_y(t)$ (gaya *thrust* pada sumbu X dan Y). Sedangkan gerakan rotasi $\dot{r}(t)$ dihasilkan dari gaya *drag* $D_z(t)$.

Gaya *thrust* merupakan gaya angkat *quadcopter* yang dipengaruhi oleh konstanta *thrust* b , panjang lengan *quadcopter* l , dan kecepatan putar setiap motor Ω_n yang dapat dituliskan persamaannya sebagai berikut:

$$\tau_x(t) = bl(\Omega_4^2 - \Omega_2^2)(t) \quad (7)$$

$$\tau_y(t) = bl(\Omega_3^2 - \Omega_1^2)(t) \quad (8)$$

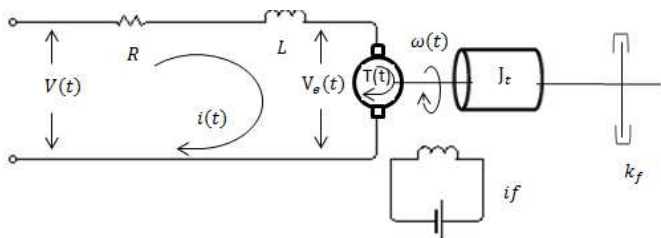
$$\tau_z(t) = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)(t) \quad (9)$$

Gaya *drag* merupakan gaya hambat yang dihasilkan oleh gesekan baling-baling terhadap udara yang dapat dituliskan persamaannya sebagai (10).

$$D_z(t) = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)(t) \quad (10)$$

C. Model Motor

Karena *input* setiap pergerakan *quadcopter* merupakan kecepatan putar motor, maka motornya juga perlu dimodelkan. Motor *brushless* DC dapat dimodelkan seperti model motor *brushed* DC [4] [6] dengan gambar skematiknya seperti terlihat pada Gbr 4.



Gbr. 4. Diagram skematik motor

Komponen elektrik dasar motor terdiri dari arus $i(t)$ yang mengalir melewati resistansi R dan induktansi L pada setiap kumparan sehingga menghasilkan medan magnet if , dengan *input* tegangan $V(t)$ dan *output* tegangan emf balik $V_e(t)$ yang dapat dituliskan sebagai (11).

$$V(t) = Ri(t) + L \frac{di(t)}{dt} + V_e(t) \quad (11)$$

Tegangan emf balik $V_e(t)$ proporsional terhadap kecepatan sudut $\omega(t)$ yang menghasilkan konstanta emf balik K_b yang dapat dituliskan sebagai (12).

$$V_e(t) = K_b \omega(t) \quad (12)$$

Torsi $T(t)$ yang proporsional terhadap arus $i(t)$ menghasilkan konstanta motor K_m yang dapat dituliskan sebagai (13).

$$T(t) = K_m i(t) \quad (13)$$

Besar torsi $T(t)$ juga bergantung pada momen inersia motor J_t dan koefisien gesek k_f sehingga persamaan mekanik motor dapat dituliskan sebagai (14).

$$J_t \frac{d\omega(t)}{dt} = T(t) - k_f \omega(t) \quad (14)$$

Jika persamaan motor (11) – (14) digabungkan dan diubah ke bentuk Laplace dengan menganggap induktansi L dan koefisien gesek k_f bernilai 0 karena nilainya yang kecil [7], maka akan membentuk persamaan *plant* motor $G_m(s)$ (15).

$$G_m(s) = \frac{K_m}{RJs + K_m k_b} \quad (15)$$

D. Perancangan Plant

Sebelum diberi pengendali, persamaan *quadcopter* (1) – (6) yang telah didapatkan harus dilinearisasikan. Linearisasi dilakukan berdasarkan *Taylor series*. Setelah dilinearisasikan dan ditentukan nilai *equilibrium* berdasarkan posisi *hover*, maka didapatkan persamaan berikut:

$$\delta \dot{u}(t) = -g \delta \theta(t) \quad (16)$$

$$\delta \dot{v}(t) = g \delta \phi(t) \quad (17)$$

$$\delta \dot{w}(t) = \frac{1}{m} \delta \tau_z(t) \quad (18)$$

$$\delta \dot{p}(t) = \frac{1}{I_x} \delta \tau_x(t) \quad (19)$$

$$\delta \dot{q}(t) = \frac{1}{I_y} \delta \tau_y(t) \quad (20)$$

$$\delta \dot{r}(t) = \frac{1}{I_z} \delta D_z(t) \quad (21)$$

Karena pengendali yang dirancang pada penelitian ini hanya terbatas pada gerakan *hover* dan *roll*, maka persamaan yang digunakan hanya persamaan *hover* (18) dan persamaan *Roll* (19). Bentuk Laplace dari persamaan tersebut adalah:

$$sW(s) = \frac{1}{m} \tau_z(s) \quad (22)$$

$$sP(s) = \frac{1}{I_x} \tau_\phi(s) \quad (23)$$

Jika persamaan (22) digabung dengan persamaan motor (15), maka akan menghasilkan persamaan *plant hover* keseluruhan $G_{al}(s)$ (24).

$$G_{al}(s) = \frac{bK_m^2}{ms(RJ_t s + K_m k_b)(RJ_t s + K_m k_b)} \quad (24)$$

Jika persamaan (23) digabung dengan persamaan motor (15), maka akan menghasilkan persamaan *plant Roll* keseluruhan $G_{roll}(s)$ (25).

$$G_{roll}(s) = \frac{p(s)}{v_2^2 - v_4^2} = \frac{bIK_m^2}{I_x s(RJ_t s + K_m k_b)(RJ_t s + K_m k_b)} \quad (25)$$

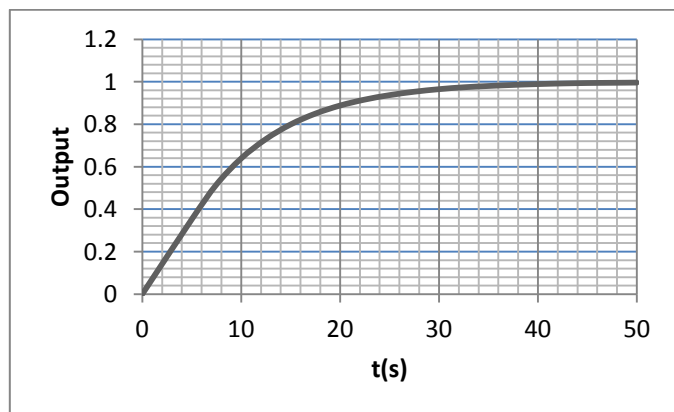
E. Identifikasi Quadcopter

Quadcopter yang diteliti memiliki spesifikasi yang dapat dilihat pada TABEL I.

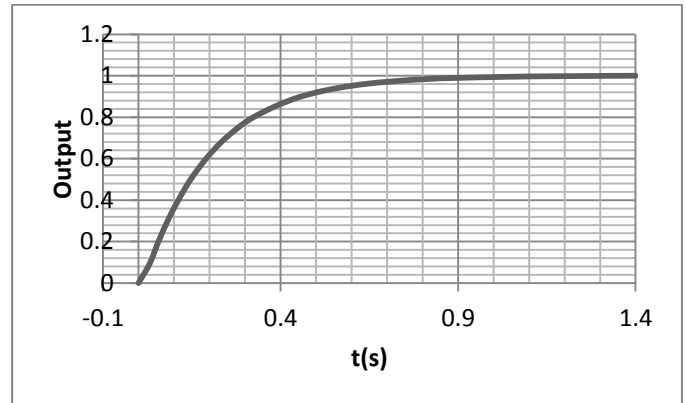
TABEL II
DATA HASIL PENGUKURAN

Simbol	Nilai	Keterangan
b	$1,1732 \cdot 10^{-7}$	Konstanta <i>thrust</i> motor
l	0,163 m	Panjang lengan <i>quadcopter</i>
I_x	$4,00103 \cdot 10^{-3} kg \cdot m^2$	Momen inersia pada sumbu X
I_y	$4,00103 \cdot 10^{-3} kg \cdot m^2$	Momen inersia pada sumbu Y
R	0,12 Ω	Resistansi motor
J_t	$4,6475 \cdot 10^{-6} kg \cdot m^2$	Momen inersia motor
K_m	$1 \cdot 10^{-3}$	Konstanta motor
k_b	$9,9314 \cdot 10^{-4}$	Konstanta emf balik motor
m	1,163 kg	Massa <i>quadcopter</i>

Setelah seluruh data spesifikasi *quadcopter* telah diukur, maka jika dimasukkan ke dalam persamaan *hover* (24) dan persamaan *roll* (25), akan menghasilkan grafik respon *transient* seperti yang terlihat pada Gbr. 5 dan Gbr. 6.



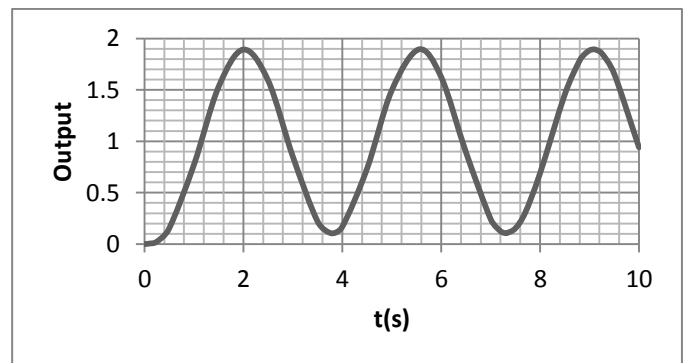
Gbr. 5. Grafik respon *plant hover* $G_{al}(s)$ tanpa pengendali



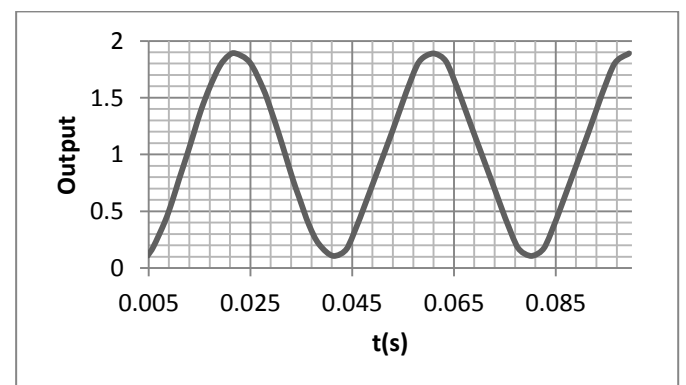
Gbr. 6. Grafik respon *plant roll* $G_{roll}(s)$ tanpa pengendali

F. Perancangan Pengendali

Untuk menerapkan metode Ziegler-Nichols dan Tyreus-Luyben, perlu didapatkan nilai *critical proportional gain* K_{cr} yang merupakan nilai penguat terbesar yang menghasilkan grafik osilasi seperti terlihat pada Gbr. 7 dan Gbr.8.



Gbr. 7. Grafik osilasi $G_{al}(s)$ dengan $K_{cr} = 34,82032017$



Gbr. 8. Grafik osilasi $G_{roll}(s)$ dengan $K_{cr} = 67,11547084$

Dengan menggunakan kestabilan Routh, didapatkan nilai K_{cr} sebesar 34,82032017 untuk persamaan *hover* dan K_{cr} sebesar 67,11547084 untuk persamaan *roll*. Dari grafik osilasi tersebut didapatkan nilai jarak antar dua gelombang P_{cr} sebesar 3,56 untuk persamaan *hover* dan P_{cr} sebesar 0,04 untuk persamaan *roll*

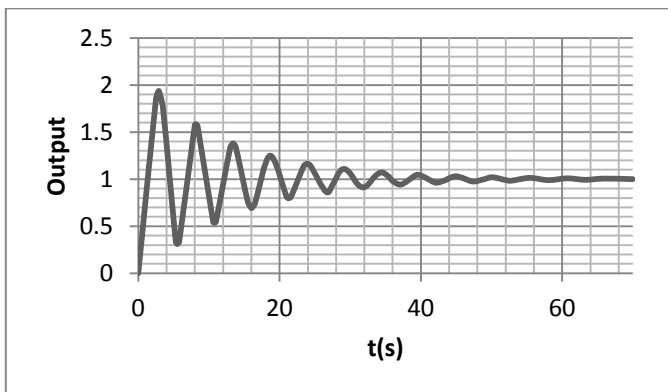
TABEL II
 ATURAN TUNING METODE ZIEGLER-NICHOLS [8]

Pengendali	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Sehingga, berdasarkan TABEL II yang merupakan hasil dari penelitian yang telah dilakukan oleh Ziegler dan Nichols mengenai *tuning* nilai PID, didapatkan persamaan pengendali $G_c(s)$ (26).

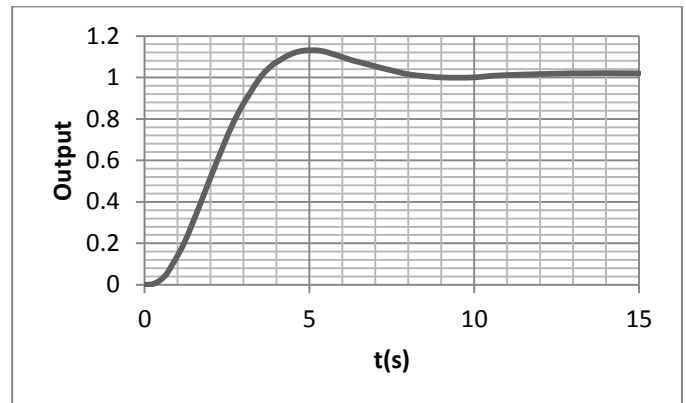
$$G_c(s) = \frac{15,66914408s + 5,281733959}{s} \quad (26)$$

Pengendali PI (26) yang telah didapat akan mengendalikan *plant hover* $G_{al}(s)$ sehingga menghasilkan grafik seperti yang terlihat pada Gbr. 9.



Gbr. 9. Grafik *plant hover* $G_{al}(s)$ dengan nilai PI yang didapatkan dari aturan *tuning* Ziegler-Nichols

Karena grafik pada Gbr. 9 menunjukkan osilasi dan *overshoot* yang begitu besar (melebihi 25%), maka perlu dilakukan *tuning* tambahan [8]. Namun berdasarkan spesifikasi motor yang digunakan, *overshoot* terbesar yang diperbolehkan hanya berkisar 16%. *Tuning* tambahan dilakukan dengan cara menurunkan nilai *proportional gain* K_p dan *integral gain* K_i . Setelah dilakukan *tuning* tambahan, didapatkan grafik respon *transient* pada Gbr. 10.



Gbr. 10. Grafik *plant hover* $G_{al}(s)$ dengan nilai K_p sebesar 5,66914408 dan K_i sebesar 0,05281733959 yang didapatkan dari *tuning* tambahan

Dari *tuning* tambahan yang telah dilakukan, didapatkan grafik dengan *overshoot* yang kurang dari 16% dengan nilai *proportional gain* K_p sebesar 5,66914408 dan *integral gain* K_i sebesar 0,05281733959.

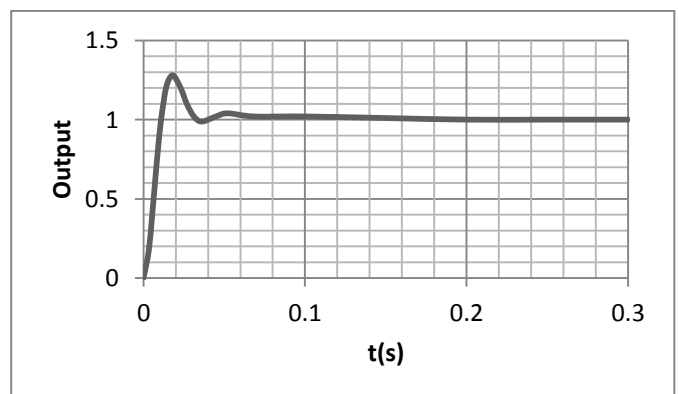
TABEL III
 TABEL PENGATURAN PID TYREUS-LUYBEN

Controller	K_p	τ_I	τ_D
PI	$K_{cr}/3.2$	$2.2P_{cr}$	—
PID	$K_{cr}/2.2$	$2.2P_{cr}$	$P_{cr}/6.3$

Berdasarkan TABEL III yang merupakan tabel pengaturan PID Tyreus-Luyben, didapatkan persamaan pengendali $G_{roll}(s)$ (27).

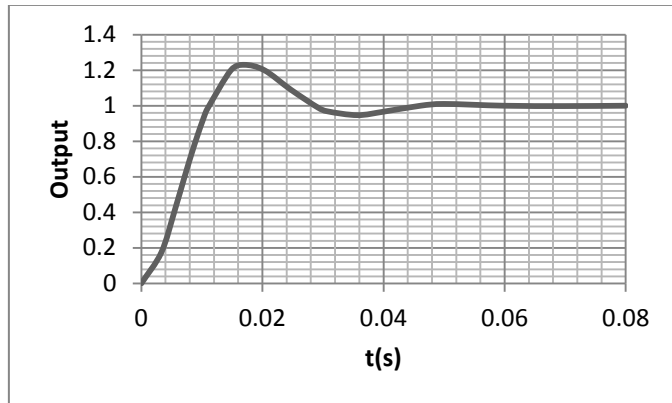
$$G_c(s) = \frac{0.426113135s^2 + 67,11547084s + 762.67045}{s} \quad (27)$$

Dengan begitu hasil grafik dari *plant quadcopter* dengan menggunakan control PID Tyreus-Luyben adalah



Gbr. 11 Respon sinyal uji *step* untuk gerakan *roll*

Dikarena untuk mencapai steady state masih lama, maka dilakukan tuning terhadap nilai K_i , sehingga grafiknya menjadi seperti pada gambar 5, grafik ini lebih cepat mencapai kestabilan dibandingkan dengan penelitian sebelumnya[2]



Gbr. 12 Respon sinyal uji *step* untuk gerakan *roll* setelah melakukan *tuning* terhadap nilai K_i

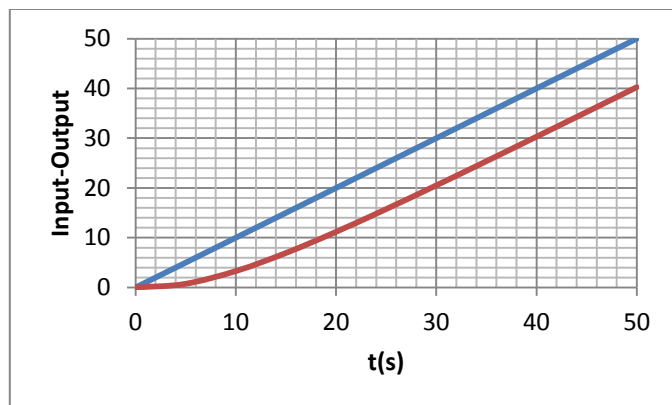
Dengan demikian, nilai PID untuk persamaan *roll* adalah $K_p = 67,11547084$, $K_i = 7,6267045$, $K_d = 0.426113135$

V. HASIL DAN ANALISIS

Nilai pengendali yang telah didapatkan akan dianalisis langsung dengan cara pengujian. Namun, sebelum dilakukan pengujian, perlu dilakukan analisis terhadap *error* agar dapat diketahui apakah pengendali telah mampu memperkecil *error* atau tidak.

G. Analisis Error

Untuk menganalisis *error hover*, perlu dibuatkan respon sistem terhadap sinyal uji *ramp*.



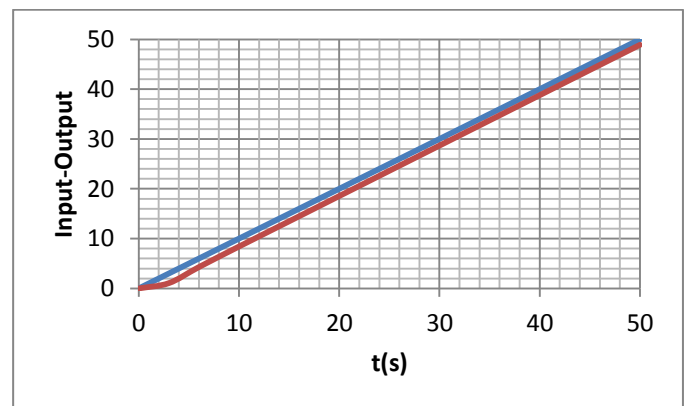
Gbr. 13. Grafik *error* $G_{ai}(s)$ tanpa pengendali

Gbr. 13 menunjukkan respon sistem tanpa pengendali terhadap sinyal uji *ramp*. Dari grafik tersebut didapatkan data *error* seperti yang terlihat pada TABEL IV.

TABEL IV
 DATA LUAS ERROR PADA *PLANT HOVER* $G_{ai}(s)$ TANPA PENGENDALI

<i>Input</i>	<i>Output</i>	Luas <i>error</i> e_{ss} <i>Input – Output</i>
0	0	0
5	0,7502	4,2498
10	3,299	6,701
15	6,935	8,065
20	11,18	8,82
25	15,75	9,25
30	20,51	9,49
35	25,39	9,61
40	30,31	9,69
45	35,27	9,73
50	40,25	9,75

Selanjutnya akan dibuat grafik dari *error plant* yang menggunakan yang dapat dilihat pada Gbr. 14.



Gbr. 14. Grafik *error* $G_{ai}(s)$ dengan pengendali PI

Gbr. 14 menunjukkan respon sistem tanpa pengendali terhadap sinyal uji *ramp*. Dari grafik tersebut didapatkan data *error* seperti yang terlihat pada TABEL V.

TABEL V
 DATA LUAS ERROR PADA *PLANT HOVER* $G_{ai}(s)$ DENGAN PENGENDALI PI

<i>Input</i>	<i>Output</i>	Luas <i>error</i> e_{ss} <i>Input – Output</i>
0	0	0
3	1,063	1,937
6	4,283	1,717

Prosiding
ANNUAL RESEARCH SEMINAR 2016
 6 Desember 2016, Vol 2 No. 1

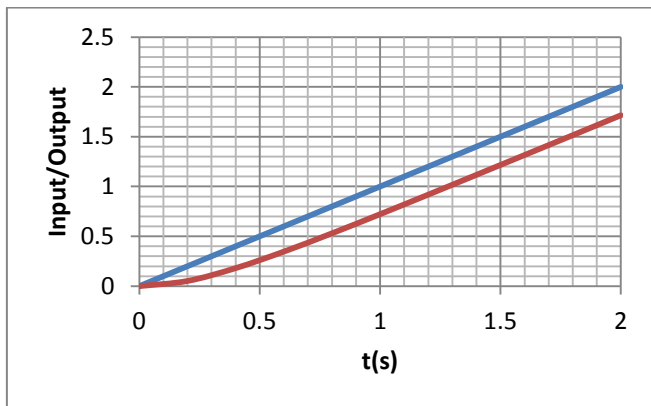
ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

10	8,409	1,591
15	13,47	1,53
20	18,54	1,46
25	23,61	1,39
30	28,67	1,33
35	33,73	1,27
40	38,79	1,21
45	43,85	1,15
50	48,9	1,1

Dari data yang diperoleh dari TABEL V, didapatkan luas *error* pada saat sistem mencapai *steady-state* (*time* 15) memiliki luas *error* sebesar 1,53. Jika dibandingkan dengan nilai luas *error* pada *plant hover* tanpa pengendali (TABEL IV) yang memiliki luas *error* pada saat mencapai *steady-state* (*time* 50) sebesar 9,75, dapat dikatakan bahwa pengendali PI telah berhasil menurunkan nilai *error* yang terdapat pada sistem.

Untuk menganalisis *error roll*, perlu dibuatkan respon sistem terhadap sinyal uji *ramp*.



Gbr. 15. Grafik error $G_{roll}(s)$ dengan pengendali PID

Pada tabel VI menunjukkan nilai luas kesalahan yang diperoleh dalam keadaan *steady state* sebesar 0,285, nilai tersebut diperoleh dari grafik *error* pada Gbr. 15

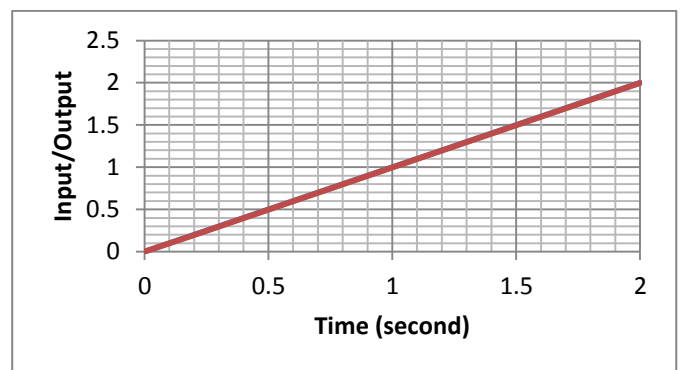
TABEL VI

TABEL LUAS KESALAHAN PADA PLANT QUADCOPTER

Input	Output	Luas error e_{ss} $Input - Output$
0	0	0
0.2	0,05194	0,14806
0.4	0,1807	0,2193
0.6	0,3465	0,2535
0.8	0,5301	0,2699
1	0,7223	0,2777

1.2	0,9185	0,2815
1.4	1,117	0,283
1.6	1,316	0,284
1.8	1,515	0,285
2	1,715	0,285

Pada Gbr. 16 menunjukkan grafik *error* pada plant *quadcopter* dengan controller PID, dimana garis berwarna merah mendekati garis yang berwarna biru yang merupakan nilai *input*.



Gbr. 16. Grafik error $G_{roll}(s)$ dengan pengendali PID

Pada tabel VII menunjukkan nilai luas kesalahan yang diperoleh dalam keadaan *steady state* sebesar 0,003, nilai tersebut diperoleh dari grafik *error* pada Gbr. 15.

TABEL VII

TABEL LUAS KESALAHAN PADA PLANT QUADCOPTER DENGAN CONTROLLER PID

Input	Output	Luas error e_{ss} $Input - Output$
0	0	0
0.2	0.1958	0,0042
0.4	0.3959	0,0041
0.6	0.596	0,004
0.8	0.7961	0,0039
1.2	1.196	0,004
1.4	1.396	0,004
1.6	1.596	0,004
1.8	1.797	0,003
2	1.997	0,003

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

III. KESIMPULAN

Dari penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Perancangan pengendali PI pada *plant hover* dengan menggunakan metode Ziegler-Nichols membutuhkan *tuning* tambahan agar mencapai *overshoot* 16%.
2. Pada *tuning* tambahan, nilai *proportional gain* K_p dan *integral gain* K_i hanya perlu diturunkan dari nilai *starting point* yang didapatkan dari metode Ziegler-Nichols agar mencapai respon terbaik.
3. Nilai *proportional gain* K_p yang didapat dari metode Ziegler-Nichols sebesar 15,66914408, dan *integral gain* K_i sebesar 5,281733959. Sedangkan nilai *proportional gain* K_p terbaik agar mencapai grafik maksimum *overshoot* kurang dari 16% yang didapat sebesar 5,66914408 dan *integral gain* K_i sebesar 0,05281733959.
4. Dengan menurunkan nilai *proportional gain* K_p , dapat menurunkan *overshoot*, dan meningkatkan *raise time*, dan dengan menurunkan nilai *integral gain* K_i , dapat menurunkan *overshoot* dan *settling time*.
5. *Steady-state error* dapat dikurangi dengan menambahkan *integral gain* K_i . Dan osilasi dapat dikurangi dengan mengurangi *proportional gain* K_p dan *integral gain* K_i .
6. Metode Ziegler-Nichols dan Tyreus-Luyben secara simulasi dapat diimplementasikan pada *plant quadcopter*.
7. Respon dan *steady state* sinyal uji *step* terhadap *plant quadcopter* lebih cepat jika menggunakan metode PID Tyreus-Luyben, jika dibandingkan dengan respon dan *steady state* tanpa *control*.
8. Metode Tyreus-Luyben dapat diimplementasikan kepada *quadcopter* hanya sebagai *starting point* PID.
9. Nilai PID yang diperoleh adalah
 $K_p = 67,11547084$
 $K_i = 7,6267045$
 $K_d = 0.426113135$

REFERENSI

- [1] T. Aw, K. As, and R. Hakim, "Perancangan Dan Implementasi Kontroler Gain Scheduling Untuk Gerakan Lateral Way-To-Way Point Pada Uav Quadcopter," pp. 1–5.
- [2] K. H. Ang, G. Chong, S. Member, and Y. Li, "PID Control System Analysis, Design, and Technology," *IEEE*, vol. 13, no. 4, pp. 559–576, 2005.
- [3] Z. He and L. Zhao, "A Simple Attitude Control of Quadrotor Helicopter Based on Ziegler-Nichols Rules for Tuning PD Parameters," *Hindawi*, vol. 2014, 2014.
- [4] M. De Oliveira, "Modeling, Identification and Control of a Quadrotor Aircraft," Czech Technical, 2011.
- [5] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation*, 3rd ed. Hoboken: John Wiley, 2016.
- [6] S. Sekalala, "Performance of a Three-Phase Permanent Magnet Motor Operating as a Synchronous Motor and a Brushless DC Motor," Louisiana State University, 2006.
- [7] O. J. Oguntoyinbo, "PID Control Of Brushless Dc Motor And Robot Trajectory Planning And Simulation With Matlab/Simulink," 2009.
- [8] K. Ogata, *Modern Control Engineering*, 5th ed. New Jersey: pearson, 2010.